What is NP?
- Interpretation of a Chinese paradox *white horse is not horse*

Yu LI

(1) MIS, Université de Picardie Jules Verne, 33 rue Saint-Leu, 80090 Amiens, France
(2) Institut of computational theory and application, Huazhong University of Science and Technology, Wuhan, China

Abstract

The notion of *nondeterminism* has disappeared from the current definition of NP, which has led to ambiguities in understanding NP, and caused fundamental difficulties in studying the relation *P versus NP*. In this paper, we question the equivalence of the two definitions of NP, the one defining NP as the class of problems solvable by a nondeterministic Turing machine in polynomial time, and the other defining NP as the class of problems verifiable by a deterministic Turing machine in polynomial time, and reveal cognitive biases in this equivalence. Inspired from a famous Chinese paradox *white horse is not horse*, we further analyze these cognitive biases. The work shows that these cognitive biases arise from the confusion between different levels of *nondeterminism* and *determinism*, due to the lack of understanding about the essence of *nondeterminism*. Therefore, we argue that fundamental difficulties in understanding *P versus NP* lie firstly at cognition level, then logic level.

Keywords: recognition of problem; P versus NP; nondeterminism; determinism; polynomial time verifiability; nondeterministic Turing machine; deterministic Turing machine; *white horse is not horse*

1 Introduction

One of the main issues in the theory of NP-completeness is to study whether a NP problem can be efficiently solved by an algorithm, i.e., the existence of a polynomial time algorithm to solve a NP problem.

However, not only does the general public find it difficult to understand the theory of NP-completeness, but also a large number of computer scientists and students feel it half-comprehended. In our opinion, a major cause for this phenomenon is that there exist ambiguities in understanding NP. Initially, NP refers to *Nondeterministic Polynomial time*, but the notion of *nondeterminism* is lost in the current definition of NP. Consequently, although one can give out the definition of

Email address: yu.li@u-picardie.fr (Yu LI).

NP as in the literature, but one cannot give a satisfactory explanation about what NP is. Furthermore, P *versus* NP has always remained one of the most perplexing problems in the theory of computational complexity since it was raised in 1971 [1].

In this paper, we investigate the recognition of NP by questioning the equivalence of the two definitions of NP.

The paper is organized as follows. In Section 2, we carry out a brief survey of the definition of NP. In Section 3, we make a preliminary discussion about the recognition of *problem* with the help of the etymology in English and in Chinese. In Section 4, we interpret two well-known arguments for supporting the equivalence of the two definitions of NP. In Section 5, we make use of a famous Chinese paradox *white horse is not horse* to further analyze the cognitive biases in this equivalence. In Section 6, we conclude the paper.

2 Overview of the definition of NP

Concerning the definition of NP, it can be traced to the 60's [2][3], where a great number of applicable and significant problems were discovered for which no polynomial algorithms could be found to solve them. As time passed and much effort was expended on attempts at efficiently solving these problems, it began to be suspected that there was no such solution.

Then, the theory of computational complexity was born in order to study these problems. The class of these problems was defined as problems solvable by nondeterministic Turing machine in polynomial time, denoted as NP (*Nondeterministic Polynomial time*), to distinguish from those known problems solvable by deterministic Turing machine in polynomial time, denoted as P (*Polynomial time*).

Cook's theorem, presented in a paper entitled *The complexity of theorem proving procedures* [1] in 1971, laid the foundation for the theory of NP-completeness that becomes the core of the current theory of computational complexity. Since then, there exist two definitions for NP:

- **Solvability-based definition:** NP is the class of problems solvable by a nondeterministic Turing machine in polynomial time.
- **Verifiability-based definition:** NP is the class of problems verifiable by a deterministic Turing machine in polynomial time.

In order to facilitate the following discussion, we denote the class defined by the verifiability-based definition as $P_{verifiable}$, and keep NP to denote the class defined by the solvability-based definition.

The current academic community generally considers the two above definitions to be equivalent [4], that is, $NP = P_{verifiable}$:

The two definitions of NP as the class of problems solvable by a nondeterministic Turing machine in polynomial time and the class of problems verifiable by a deterministic Turing machine in polynomial time are equivalent. The proof is described by many textbooks, for example Sipser's Introduction to the Theory of Computation, section 7.3 [5].

	$P \neq NP$	$P = NP$	*Ind*	*DC*	*DK*	*DK and DC*	*Other*
2002	61(61%)	9(9%)	4(4%)	1(1%)	22(22%)	0(0%)	3(3%)
2012	126(83%)	12(9%)	5(3%)	5(3%)	1(0.6%)	1(0.6%)	1(0.6%)

Table 1: Result of two polls about the opinions about *P versus NP*, where *Ind* stands for Independent of ZFC. *DC* stands for Do not Care. *DK* stands for Do not Know.

Therefore, the verifiability-based definition is used to replace the solvability-based one, and has become the standard definition of NP:

P is the class of problems solvable by a deterministic Turing machine in polynomial time, while NP is the class of problems verifiable by a deterministic Turing machine in polynomial time.

Clearly, $P \subseteq NP$, since for each problem in P its solutions can be verified in polynomial time by a deterministic Turing machine.

Furthermore, the famous problem of *P versus NP* consists in asking whether a problem verifiable by a polynomial algorithm is solvable by a polynomial algorithm, denoted as $NP = P$?

Such a problem has become a major unsolved problem in the theory of computational complexity [3][6][7][8], and was selected as one of the seven Millennium Prize Problems by the Clay Mathematics Institute [9]. Although a lot of effort has been done (see a comprehensive list maintained by Woeginger (2010) [10]), no substantial progress has been achieved until now.

William Gasarch conducted two polls about the future of *P versus NP* [11], with 100 participants in 2002 and 152 participants in 2012. The result (see Table 1) shows that:

There is a definite trend - more people think $P \neq NP$ now than they did in 2002. How strongly held are these opinions? Although I did not ask people what the strength of their opinion was in either poll (1) in 2002, 7 out of the 61 $P \neq NP$ votes (11%) said they had some doubts that $P \neq NP$, (2) in 2011, 16 out of the 125 (again 11%) said they had some doubts that $P \neq NP$. Hence, of the people that think $P \neq NP$, the level of confidence is about the same.

In our opinion, the difficulty in understanding *P versus NP* lies firstly at cognition level, then logic level. In fact, due to the fact that the verifiability-based definition has been accepted as the standard definition of NP, the notion of *nondeterminism* has disappeared from NP, which has caused ambiguities in understanding NP. So if we hope to get an insight into *P versus NP*, we need at first to question the equivalence of the two definitions of NP.

In the following, we start from a preliminary discussion about the recognition of problem, then we interpret two well-known arguments for supporting $NP = P_{verifiable}$.

3 Preliminary discussion about the recognition of problem

Problem is the center of thought. Around a problem, all ideas flourish. So without a problem, there would be no thought. However, if one wants to ask some general questions about *problem*

such as, *where does a problem come from? what is the essence of a problem? how to define a problem?* one would find it difficult to answer these questions, because such an investigation is absent from the traditional philosophy.

Starting from the etymology of *problem* in English and in Chinese, we make a preliminary discussion about the recognition of problem.

3.1 Etymology of *problem*

3.1.1 In English

In Greek, proballo = pro (in face of) + ballo (to throw).

Problem means an obstacle to prevent from reaching a goal.

3.1.2 In Chinese

In Chinese, *problem* consists of two characters, and each character represents an image that is composed by several basic characters [12][13]:

问题＝问＋题: problem
问＝门(door) ＋口(mouth)：ask for
题＝是(affirmation) ＋页(head)：important subject

Then, the Chinese characters of *problem* describe a scene where a person outside a door knocks and asks for a question, and a person inside the door opens and answers the question.

Therefore, *problem* refers to an interface between the questioner and the responder, which means questions about the essence of things.

3.1.3 Interpretation of *problem* in the context of *problem-solving*

In the context of *problem-solving*, *problem* can be interpreted in two ways inspired by the above etymology.

When a problem has been recognized out, it can be clearly defined by specifying the input data and the output result, and solved by designing an algorithm. In this case, such a problem is not really problematique.

When a problem has not been recognized out, it is obscure, so it needs to question about the essence of things in order to recognize it. In this case, such a problem becomes problematique.

The two interpretations are complementary and both contribute to *problem-solving*.

3.2 Recognition of NP

From the view of cognition, *recognition* is *cognition* again, that is, *recognition* means to distingush something new from some previously known things, rather than identify it as those known things.

Such a principe of recognition can be stated as *Yin-Yang principe* in the traditional Chinese thought. We refer to a text from *Dao De Jing*, a Chinese classic, to clarify this idea [14]:

When the people of the world all know beauty as beauty, there arises the recognition of ugliness.

When they all know the good as good, there arises the recognition of bad.

Therefore being and non-being produce each other; difficult and easy complete each other; long and short contrast each other;

High and low distinguish each other; sound and voice harmonize with each other; beginning and end follow each other.

Concerning the recognition of NP, the key is to determine the property that can distinguish NP from P.

The concept of NP (*Nondeterministic Polynomial time*) is defined in terms of nondeterministic Turing machine, that is, NP is defined at *algorithm* level rather than *problem* level. NP implies the two properties, the *nondeterminism* referring to the multiple choices of transition, and the *polynomial time verifiability* referring to the verifiability of solution in polynomial time.

About the concept of *nondeterminism*, it is under investigation, so the definition at algorithm level is different from that at problem level in the view of cognition, as discussed in the above two interpretations of *problem*. Therefore, the *nondeterminism* in terms of the multiple choices of transition just expresses the formal sense of nondeterminism at algorithm level, but not the essential sense of nondeterminism at problem level. In order to continue our discussion, we temporarily use such nondeterminism as denotation of concept to recognize NP from P. For the concept of *polynomial time verifiability*, it concerns the *determinism* that is already clear in the view of cognition.

The equivalence of the two definitions of NP suggests that the property of *Polynomial time verifiability* is used to recognize NP. Let us interpret two well-known arguments for supporting this equivalence.

4 Interpretation of two well-known arguments for $NP = P_{verifiable}$

4.1 A popular argument for $NP = P_{verifiable}$

A popular argument accepted by the general public can be stated as:

Because the solution of a NP problem is verifiable in polynomial time, so NP is the class of problems verifiable in polynomial time ($NP = P_{verifiable}$).

The argument consists of two basic propositions. The first, *the solution of a NP problem is verifiable in polynomial time*, means that a NP problem is a member of the set $P_{verifiable}$ and possesses the property of polynomial time verifiability. The second, *NP is the class of problems verifiable in polynomial time*, means that the set NP is identified as the set $P_{verifiable}$, i.e., $NP = P_{verifiable}$. The argument uses the polynomial time verifiability that defines $P_{verifiable}$ to recognize NP as $P_{verifiable}$.

Although a NP problem is verifiable in polynomial time at logic level, but this property is not able to recognize NP from P at cognition level, because the essence of this property is the *determinism* and it is shared by P, since a P problem is verifiable in polynomial time. Therefore, this argument is not valid. In other words, if the polynomial time verifiability is used to define NP, the *nondeterminism* would disappear at cognition level, as the *determinism* and the *nondeterminism* are relative at the same level of concept.

4.2 A professional argument for $NP = P_{verifiable}$

A professional argument accepted in the theory of NP-completeness can be stated as:

Because a deterministic Turing machine can be considered as a special case of a nondeterministic Turing machine, so $P \subseteq NP$ thus $NP = P_{verifiable}$.

In fact, the first proposition, *a deterministic Turing machine can be considered as a special case of a nondeterministic Turing machine*, is a pure mathematical deduction, that is, *if $(n = 1)$ and $(n > 1)$ then $(n \geq 1)$*, where $(n = 1)$ refers to only one transition, $(n > 1)$ several possible transitions, and $(n \geq 1)$ one or several possible transitions. In other words, this proposition is just valid at mathematical logic level, because the sense of the transition number n is not taken into account. For the second proposition, it concerns the recognition of NP and is situated at cognition level.

Therefore, although $(n = 1)$ and $(n > 1)$ can be mixed to be $(n \geq 1)$ at mathematical logic level, but P and NP cannot be confused in terms of $P \subseteq NP$ at cognition level, because the relation $P \subseteq NP$ itself needs to be proved. This *question begging* leads to $NP = P_{verifiable}$, then the nondeterminism disappears from NP.

5 Interpretation of a Chinese paradox *white horse is not horse*

The cognitive biases in the above recognition of NP are quite typical and widespread. For example, a famous Chinese paradox called *white horse is not horse* [15], illustrates such cognitive biases.

The paradox of *white horse is not horse* was proposed by a great Chinese logician *Gongsun Long* (325 - 250 BC), and it tells a history that, one day, *Gongsun Long* went to a city on riding a white horse. At the gate of the city, the guard said to him that, *your white horse is horse*, in according to regulations, horses were not allowed to enter the city. Thus, *Gongsun Long* began his argument - *white horse is not horse*, finally he persuaded the guard, and entered the city riding on his white horse.

The argument of *Gongsun Long* was presented in his writing known as *White Horse Dialogue* between two speakers. Here is a partial summary:

The issue : Can it be that white horse is not horse?

Advocate: It can.

Objector: How so?

Advocate: Horse is named by means of the shape. White is named by means of the color. What names the color is not what names the shape. Hence, I say that White horse is not Horse.

Objector: Having a white horse cannot be said as having no horses. As it cannot be said as having no horses, it means having a horse. Having a white horse is having a horse, how can a white one not be horse?

Advocate: If one wants a horse, that extends to yellow or black horses. But if one wants a white horse, that does not extend to yellow or black horses. Suppose that a white horse were horse, then what one wants would be horse. If what one wants were horse, then a white horse would not differ from a horse. If what one wants does not differ, then how is it that a yellow or black horse is sometimes acceptable and sometimes unacceptable? It is clear that acceptable and unacceptable are mutually contrary. Hence, given a yellow or black horse, one can respond that there are horses, but one cannot respond that there are white horses. Thus, it is evident that white horse is not horse.

We analyze this history. For the guard, his responsibility was to recognize horses and to prevent any horse from entering the city, whether it was a white, yellow or black horse. So when he said that *your white horse is horse*, in fact he meant that *the white horse of Gongsun Long is a member of the set of horses*. However, for Gongsun Long, his objective was to recognize the set of white horses, that is, he wanted to distinguish the set of white horses with the set of yellow or black horses, so the term of *white horse* in his argument *white horse is not horse* refers to the set of white horses instead of his white horse.

The proposition of the guard and that of Gongsun Long are right respectively. But if they are confused, there would arise cognitive biases, such as saying *because a white horse is horse, so white horse is horse*, because the two propositions situate at two different levels. In the writing of *White Horse Dialogue*, Gongsun Long described such cognitive bias as a bird flying in water.

Gongsun Long just took advantage of such cognitive bias that was not perceived by the guard, and confuse the guard, so that the guard forgot his position and slided to the position of Gongsun Long, finally let the white horse of Gongsun Long enter the city.

If we make an analogy between the paradox *white horse is not horse* and the issue of the recognition of NP, we can clearly see their similarity.

6 Conclusion

We question the equivalence of the two definitions of NP and argue that there exist cognitive biases in this equivalence that arise from the confusion of levels of *nondeterminism* and *determinism*, due to the lack of understanding about the essence of *nondeterminism*. Rightly in this sense, we argue that the difficulty in understanding *P versus NP* lies firstly at cognition level, then logic level.

This work is the first step of our work of the interpretion of the theory of NP-completeness [16], and undertaken by another work of the interpretation of Cook's theorem [17]. Moreover, it provides an interpretation of a deeper work discussed in [18][19].

We hope that this work can evoke reflections from different angles about some fundamental problems in cognitive science, and contribute to understand *P versus NP*. Furthermore, we hope that this work can help to understand the complementarity of Chinese thought and Western philosophy.

Acknowledgement

This work has been guided and discussed with my masters Mr JianMing ZHOU, Mr BangFu ZHU, Mr WenQi HUANG and Mr Jacques CARLIER. It is also the fruit of the exchange with my colleagues.

References

[1] Stephen Cook, The complexity of theorem proving procedures. Proceedings of the Third Annual ACM Symposium on Theory of Computing. pp. 151-158 (1971).

[2] Fortnow Lance, Homer Steven, A Short History of Computational Complexity. Bulletin of the EATCS 80: 95133, 2003.

[3] Garey Michael R., David S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and company (1979).

[4] http://en.wikipedia.org/wiki/NP_(complexity)

[5] Michael Sipser, Introduction to the Theory of Computation, Second Edition. International Edition (2006).

[6] Martin D. Davis Ron Sigal, Elaine J. Weyuker: Computability, Complexity, and language. Academic Press In (1983).

[7] WenQi Huang, RuChu Xu, Background, Perspective and Resolution of NP-hard Problems (in Chinese). Science Edition, 2004.

[8] Scott Aaronson, Why Philosophers Should Care About Computational Complexity, Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 108 (2011).

[9] Stephen Cook, The P versus NP Problem. Clay Mathematics Institute.
http://www.claymath.org/millennium/P_vs_NP/pvsnp.pdf

[10] http://www.win.tue.nl/~gwoegi/P-versus-NP.htm

[11] William I. Gasarch, The P=?NP poll. SIGACT News Complexity Theory Column 74.
http://www.cs.umd.edu/~gasarch/papers/poll2012.pdf

[12] Xu Shen, Explaining Simple and Analyzing Compound Characters (Shuowen Jiezi) (25-220 CE).
http://www.chinaknowledge.de/Literature/Script/hanzi.html

[13] BangFu Zhu, Books of Chinese characters (in Chinese).
http://open-lit.com/bookindex.php?gbid=311
http://open-lit.com/bookindex.php?gbid=627

[14] LaoZi, Tao Te Ching. http://www.wayist.org/ttc%20compared/beck.htm

[15] http://en.wikipedia.org/wiki/When_a_white_horse_is_not_a_horse.

[16] Yu Li, Interpretation of Cook's theorem - What is NP? Proceedings of the 1st International Conference on Philosophy of Information 2013, pp. 474-481, Xian, China.

[17] JianMing Zhou, Yu Li, What is Cook's theorem? (to appear).

[18] JianMing Zhou, Algorithm and Chinese Raison. http://wenku.baidu.com/view/c2e72710f18583d0496459d7.html# (in Chinese).

[19] JianMing Zhou, Computability vs. Nondeterministic and P vs. NP. http://arxiv.org/abs/1305.4029.